

35.C13864

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: )  
HIROSHI SATOMI )  
Application No.: 09/406,721 )  
Filed: September 28, 1999 )  
For: APPARATUS AND METHOD OF )  
PROGESSING CHARACTERS : December 10, 1999

Assistant Commissioner for Patents  
Washington, D.C. 20231

CLAIM TO PRIORITY

Sir:

Applicant hereby claims priority under the  
International Convention and all rights to which he is  
entitled under 35 U.S.C. § 119 based upon Japanese Priority  
Application No. 10-275233, filed September 29, 1998.

A certified copy of the priority document is  
enclosed.

Applicant's undersigned attorney may be reached in  
our Costa Mesa, CA office by telephone at (714) 540-8700.  
All correspondence should continue to be directed to our  
below listed address.

Respectfully submitted,

  
Attorney for Applicant

Registration No. 36,171

FITZPATRICK, CELLA, HARPER & SCINTO  
30 Rockefeller Plaza  
New York, New York 10112-3801  
Facsimile: (212) 218-2200  
F512\A0010485.DOC (Rev. 6/29/98)  
C-18294



日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

1998年 9月29日

出願番号

Application Number:

平成10年特許願第275233号

出願人

Applicant (s):

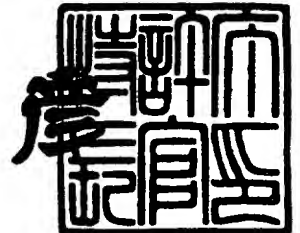
キヤノン株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

1999年10月22日

特許庁長官  
Commissioner,  
Patent Office

近藤 隆



【書類名】 特許願

【整理番号】 3828018

【提出日】 平成10年 9月29日

【あて先】 特許庁長官 伊佐山 建志 殿

【国際特許分類】 G06F 15/21

【発明の名称】 文書処理装置及び方法並びに記憶媒体

【請求項の数】 10

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社  
内

    【氏名】 里見 宏

【特許出願人】

    【識別番号】 000001007

    【住所又は居所】 東京都大田区下丸子3丁目30番2号

    【氏名又は名称】 キャノン株式会社

    【代表者】 御手洗 富士夫

    【電話番号】 03-3758-2111

【代理人】

    【識別番号】 100069877

    【住所又は居所】 東京都大田区下丸子3丁目30番2号キャノン株式会社  
内

    【弁理士】

    【氏名又は名称】 丸島 儀一

    【電話番号】 03-3758-2111

【手数料の表示】

    【予納台帳番号】 011224

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9703271

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 文書処理装置及び方法並びに記憶媒体

【特許請求の範囲】

【請求項 1】 文字データを配置する領域に関する情報を記憶する第 1 の記憶手段と、

該領域内に配置すべき文字データを該領域と対応づけて記憶する第 2 の記憶手段と、

該領域内に該文字データを一定の規則に従って配置する配置手段と、

該配置手段により該領域内に該文字データを配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断する判断手段と、

前記判断手段により、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示する表示手段と

を具備したことを特徴とする文書処理装置。

【請求項 2】 前記判断手段により全ての文字が配置できなかったと判断された場合に、配置できなかった文字データを前記第 2 の記憶手段とは別の第 3 の記憶手段に移動する移動手段を具備したことを特徴とする請求項 1 記載の文書処理装置。

【請求項 3】 前記移動手段により前記第 3 の記憶手段に移動された文字データを、前記第 1 の記憶手段に記憶されている任意の領域に対応する前記第 2 の記憶手段内の文字データ内に挿入する挿入手段を具備したことを特徴とする請求項 2 記載の文書処理装置。

【請求項 4】 前記移動手段により前記第 3 の記憶手段に移動された文字データを、前記第 1 の記憶手段に記憶されている任意の領域に対応する前記第 2 の記憶手段内の文字データと置き換える置き換え手段を具備したことを特徴とする請求項 2 記載の文書処理装置。

【請求項 5】 文字データを配置する領域に関する情報を記憶する第 1 の記憶手段と、

該領域内に配置すべき文字データを該領域と対応づけて記憶する第 2 の記憶手

段とを有する文書処理装置において、

該領域内に該文字データを一定の規則に従って配置する配置工程と、

該配置工程で該領域内に該文字データを配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断する判断工程と、

前記判断工程において、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示する表示工程と

を具備したことを特徴とする文書処理方法。

【請求項 6】 前記判断工程で全ての文字が配置できなかったと判断された場合に、配置できなかった文字データを前記第 2 の記憶手段とは別の第 3 の記憶手段に移動する移動工程を具備したことを特徴とする請求項 5 記載の文書処理方法。

【請求項 7】 前記移動工程で前記第 3 の記憶手段に移動された文字データを、前記第 1 の記憶手段に記憶されている任意の領域に対応する前記第 2 の記憶手段内の文字データ内に挿入する挿入工程を具備したことを特徴とする請求項 6 記載の文書処理方法。

【請求項 8】 前記移動工程で前記第 3 の記憶手段に移動された文字データを、前記第 1 の記憶手段に記憶されている任意の領域に対応する前記第 2 の記憶手段内の文字データと置き換える置き換え工程を具備したことを特徴とする請求項 6 記載の文書処理方法。

【請求項 9】 文字データを配置する領域に関する情報を記憶する第 1 の記憶手段と、

該領域内に配置すべき文字データを該領域と対応づけて記憶する第 2 の記憶手段とを有する文書処理装置において動作するコンピュータプログラムを格納した記憶媒体であって、

前記コンピュータプログラムは、

該領域内に該文字データを一定の規則に従って配置するための配置モジュールと、

該配置モジュールによる配置により該領域内に該文字データを配置した際に、

該領域内に、対応する全ての文字データを配置できたかどうかを判断するための判断モジュールと、

前記判断モジュールによる判断により、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示するための表示モジュールとを具備したことを特徴とする記憶媒体。

【請求項 10】 前記コンピュータプログラムは、前記判断モジュールによる判断により全ての文字が配置できなかったと判断された場合に、配置できなかった文字データを前記第 2 の記憶手段とは別の第 3 の記憶手段に移動するための移動モジュールを具備したことを特徴とする請求項 9 記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、予め定められた領域内に文字を配置する文書処理装置及び方法並びに記憶媒体に関するものである。

【0002】

【従来の技術】

近年の文字処理装置では、文書内には文字データばかりではなく、表や図形、イメージデータを混在させることのできるものが多く存在する。それらの文書処理装置では文字列を予め指定された矩形領域等一定の領域に入力したり、流し込んだりすることができるようにして、文書内に置かれた図形等と文字データが重なってしまわないように制御することができるものがある。

【0003】

また、近年Microsoft社のWindows等の普及により、文書作成を行う際に複数の文書を同時に編集することや、また、様々なアプリケーションを同時に使用することにより複数の作業を同時並行的に行うことなどが可能になってきた。

【0004】

その結果、同一の文書内に複数存在する文字データ配置用の領域内の文字データや、複数の文書間での文字データに対して、コピーや移動を容易に行えるよう

になってきた。

【0005】

【発明が解決しようとする課題】

しかしながら、複数の文書間や単一の文書内において、文字列の移動やコピー、流し込みなどが容易に行えるようになると、文書編集時に、編集者の意図していない誤操作によって、表示できる文字データ数が少ない小さな領域に大量の文字データを流し込んでしまうことがある。

【0006】

また、文字データを表示すべき矩形領域を予め指定する文書処理装置においては、指定しておいた領域のサイズや文字データの配置位置等の書式情報を、一旦文字を配置した後で変更できるのが一般的である。その際、領域のサイズや書式属性を変更する前に領域内に文字データが全て表示されていた場合でも、領域のサイズを小さくしたり文字サイズの属性を変更することによって、領域内に表示できない文字が発生する場合がある。

【0007】

そして、指定された領域内に配置しきれない文字データが発生した場合、その配置しきれなかった文字データは画面上には表示されないことになる。特に、領域内にまんべんなく文字が配置されている場合、配置しきれなかった文字データは表示されないがため、操作者は、領域から溢れ出した文字データが存在するかどうかを、視覚的に判断することができず、そのまま放置されてしまうことが多い。

【0008】

そして、このようにして放置された文字データは、表示はされないものの文字データとしては存在するために、メモリを不用意に浪費することになっていた。

【0009】

更に、配置しきれない文字データがあることに操作者が気付いた場合でも、操作者は、配置しきれなかった文字データがすべて表示されるように、文字配置領域のサイズを変更してから文書編集を行い、その後、文字配置領域のサイズを元のサイズに戻す作業を行う必要があるので、文字データの編集作業は煩雑で面倒



であった。

【0010】

また、テキストファイルを読み込んで逐次領域に流し込みを行う場合には、領域をいくら大きくしても全ての文字データを表示しきれないような場合があり、そのような場合、領域内から溢れ出た文字数ほど文字データを削除しなければならない。その際は、領域内に表示されている文字列の一部を削除して、更に、その結果表示されるようになった溢れ文字列の一部を削除するなどしなければならず、操作者は、大変煩わしい作業を行わなければならなかった。

【0011】

本発明はこのような問題点に鑑みなされたものであって、領域内に配置できない文字データがあることを操作者に認識せしめ、領域内に配置できなかった文字データを別の領域に（または同じ領域内の別の場所に）移動させることによって、領域内に配置できない文字データを容易に編集することのできる文書処理装置及び方法並びに記憶媒体を提供することを目的とする。

【0012】

【課題を解決するための手段】

上記目的を達成するために本発明に係る文書処理装置は、  
文字データを配置する領域に関する情報を記憶する第1の記憶手段と、  
該領域内に配置すべき文字データを該領域と対応づけて記憶する第2の記憶手段と、  
該領域内に該文字データを一定の規則に従って配置する配置手段と、  
該配置手段により該領域内に該文字データを配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断する判断手段と、  
前記判断手段により、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示する表示手段と  
を設けた。

【0013】

また、好ましくは、前記判断手段により全ての文字が配置できなかったと判断

された場合に、配置できなかった文字データを前記第2の記憶手段とは別の第3の記憶手段に移動する移動手段を設けた。

【0014】

また、好ましくは、前記移動手段により前記第3の記憶手段に移動された文字データを、前記第1の記憶手段に記憶されている任意の領域に対応する前記第2の記憶手段内の文字データ内に挿入する挿入手段を設けた。

【0015】

また、好ましくは、前記移動手段により前記第3の記憶手段に移動された文字データを、前記第1の記憶手段に記憶されている任意の領域に対応する前記第2の記憶手段内の文字データと置き換える置き換え手段を設けた。

【0016】

また、上記目的を達成するために本発明に係る文書処理方法は、  
文字データを配置する領域に関する情報を記憶する第1の記憶手段と、  
該領域内に配置すべき文字データを該領域と対応づけて記憶する第2の記憶手段とを有する文書処理装置において、  
該領域内に該文字データを一定の規則に従って配置する配置工程と、  
該配置工程で該領域内に該文字データを配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断する判断工程と、  
前記判断工程において、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示する表示工程と  
を設けた。

【0017】

また、好ましくは、前記判断工程で全ての文字が配置できなかったと判断された場合に、配置できなかった文字データを前記第2の記憶手段とは別の第3の記憶手段に移動する移動工程を設けた。

【0018】

また、好ましくは、前記移動工程で前記第3の記憶手段に移動された文字データを、前記第1の記憶手段に記憶されている任意の領域に対応する前記第2の記

憶手段内の文字データ内に挿入する挿入工程を設けた。

【0019】

また、好ましくは、前記移動工程で前記第3の記憶手段に移動された文字データを、前記第1の記憶手段に記憶されている任意の領域に対応する前記第2の記憶手段内の文字データと置き換える置き換え工程を設けた。

【0020】

また、上記目的を達成するために本発明に係る記憶媒体は、  
文字データを配置する領域に関する情報を記憶する第1の記憶手段と、  
該領域内に配置すべき文字データを該領域と対応づけて記憶する第2の記憶手段とを有する文書処理装置において動作するコンピュータプログラムを格納した記憶媒体であって、

前記コンピュータプログラムは、

該領域内に該文字データを一定の規則に従って配置するための配置モジュールと、

該配置モジュールによる配置により該領域内に該文字データを配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断するための判断モジュールと、

前記判断モジュールによる判断により、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示するための表示モジュールと

を有する。

【0021】

また、好ましくは、前記コンピュータプログラムは、前記判断モジュールによる判断により全ての文字が配置できなかったと判断された場合に、配置できなかった文字データを前記第2の記憶手段とは別の第3の記憶手段に移動するための移動モジュールを有する。

【0022】

【発明の実施の形態】

以下、本発明の実施の形態を、図面を用いながら説明する。

【0023】

図1は本発明の実施の一形態に係る文書処理装置の構成を示すブロック図である。

【0024】

本装置には、CRT（表示装置）1、VRAM（ビデオランダムアクセスメモリ）2、BMU（ビットムーブユニット）3、キーボード4、PD（ポインティングデバイス）5、CPU（中央制御装置）6、ROM（読み出し専用メモリ）7、RAM8、HDD（ハードディスクドライブ）9、FDD（フロッピーディスクドライブ）10及びNet-I/F（ネットワークインターフェース）11を含み、これらの構成要素は、アドレスバス、データバス及び制御バスからなるI/O（インプット/アウトプット）バス12により互いに接続されている。また、CRT（表示装置）1が、VRAM2に接続されている。

【0025】

CRT1は、編集中の文書や各種メッセージメニューなどを表示する。VRAM2は、CRT1において表示する文字及びイメージを展開すると共に記憶する。BMU3は、各構成要素間の情報の転送を制御する。キーボード4は、文書編集の際に操作者が用いる各種キーを備えている。PD5は、CRT1に表示されたアイコンなどを指し示すために用いられる。

【0026】

また、CPU6はROM7に格納された制御プログラムに基づき各構成要素を制御すると共に、後述する文字列配置制御処理を実行する。ROM7は、文書編集に係わるプログラム、エラー処理プログラム及び文字列配置処理用のプログラムを格納する。RAM8は、上記各種制御プログラムをCPU6が実行する際に、ワークエリア、エラー処理時の一時退避エリアとして用いられる。

【0027】

また、HDD9及びFDD10は、後述する文字列配置処理に関するプログラム、各種情報及びライブラリなどを保存する。Net-I/F11は、本装置がネットワークを経由して他の文書処理装置との間で情報の転送が行えるように、ネットワーク情報の制御及び診断を行なう。

【0028】

なお、上述したROM7に格納する制御プログラムを、本装置に別途接続されたハードディスク（HD）やフロッピーディスク（FD）、リムーバブルディスク（RD）（いずれも図示せず）等の記憶媒体に記憶することもできる。さらに、ネットワークにて接続された他の文書処理装置若しくはコンピュータに付随する記憶媒体に格納するようにしても良い。

【0029】

以下に、本実施の形態における文書編集処理に必要な情報について説明する。

【0030】

図2は本実施形態を実現するための文字配置領域を管理するために記憶される情報を示す図である。

【0031】

ここで、図2に示される諸情報は、HDD9又はFDD10の文字領域データ管理部に記憶されているものとする。

【0032】

2aは、編集を行う文書内に存在する文字領域の個数num\_areasを記憶する領域であり、2bは、文書内で管理されている文字領域のうち第1番目の文字領域に関する情報を記憶する領域であり、2cは、第n番目の文字領域に関する情報を記憶する領域である。2cまで、このように、文字領域データ管理部には、num\_area個の領域情報が順次記憶されている。

【0033】

そして、個々の文字領域に関する情報の中身が、2dから2jに記されている。

【0034】

2d、2eは、文字領域の基準となる座標値を記憶する領域である。本実施形態では、文字領域は全て矩形であり、基準座標値としては矩形の左上の頂点の座標値(pos\_x, pos\_y)を記憶する。また、2f、2gは文字領域の形状、サイズを記憶する領域である。本実施形態では、文字領域が矩形であるため、領域の高さarea\_height及び幅area\_widthを記憶する。また、2hは、本実施形態で示される文字領域内の文字データの持つ書式情報area\_infoを記憶する領域である。その中には領

域内の文字を表示するための書体名や書体サイズ、行間隔、行ピッチ、文字間隔等が記憶されている。

#### 【0035】

また、2iは、本実施形態で示される文字領域内に表示される文字データの情報を記憶する領域であり、その内容は、領域内に配置すべき文字データの個数total\_wordsを記憶する領域2kの他、配置すべき先頭の文字コードを記憶する領域2l、2番目の文字コードを記憶する領域2m、第n番目の文字コードを記憶する領域2nなどからなる。

#### 【0036】

また、2jは、本実施形態において配置、表示される文字データの配置位置を算出するための組版に使用される領域である。その詳細については図3に示されている。

#### 【0037】

図3は本実施形態で示される文字処理装置に配置、表示される文字列の配置位置を算出するために使用する組版情報である。

#### 【0038】

ここで、3aは、領域内に配置された行の数num\_linesを記憶する領域であり、3bは、第1行目の行情報であり、3cは、第n番目の行情報である。このように、3b以降には、3aで記憶された行数分の行の情報が記憶される。そして、それぞれの行の情報は、3d~3gに記されている。

#### 【0039】

3dは、領域内での行の行番号line\_noを記憶する領域であり、3eは、行のベースライン座標baselineを記憶する領域である。本実施形態では、行のベースラインは領域の先頭行の場合には矩形領域の上端(pos\_y)から文字サイズ情報だけ下の位置、先頭行でなければ直前行のベースライン位置から行ピッチ分だけ下の位置である。本実施形態では、これら文字サイズ情報や行ピッチなどの情報は、2hの書式情報内に記憶されている。

#### 【0040】

また、3fは、行の先頭に配置する文字コードの文字データ情報内でのオフセッ

ト（位置）start\_offsetを記憶しておく領域であり、3gは、行内に配置できた文字データの個数num\_wordsを記憶しておく領域である。本実施形態では、オフセットは領域内の先頭行では常に1、先頭行でない場合には直前行のオフセットにnum\_wordsを加算することによって求めることができる。

#### 【0041】

次に、本実施の形態における文字配置の制御方法をフローチャートを用いて説明する。

#### 【0042】

図4は本実施形態における領域内行配置処理を行うためのフローチャートである。

#### 【0043】

まず、ステップS401で、行数をカウントするための作業変数nを1に初期化する。次に、ステップS402にて、領域内での第n番目の行の行情報を作成する。作成された行情報は行情報の領域3cに格納される。作成される情報の詳細は図3で示した3d~3gである。実際の行情報の作成処理については、図5を用いて別途記述することにする。

#### 【0044】

そして、ステップS402の行情報作成処理において行の配置が行えない場合には、行の配置判定変数retに0が返される。

#### 【0045】

そこで、ステップS403にて、行の配置判定変数retが0であるか否かを判定する。

#### 【0046】

ステップS403でret = 0でないと判定された場合には、行の配置が行えたことを意味するので、その時は、ステップS404にてnをインクリメントし、ステップS402に戻って次の行の行情報の作成を行う。

#### 【0047】

また、ステップS403でret = 0と判定された場合には、領域内にこれ以上行の配置ができないので、ステップS405にて領域内の行数num\_linesにn-1をセット

して処理を終了する。この領域内の行数num\_linesは領域3aに記憶される。

【0048】

図5は本実施形態における第n番目の行の行情報の作成処理を表すフローチャートである。

【0049】

まず、ステップS501にて、行番号の記憶領域3dに、これから作成する行の行番号nをセットする。次に、ステップS502にて、 $n = 1$ か否かを判定する。

【0050】

ステップS502で $n = 1$ と判定された場合には、作成する行情報は領域内での先頭行の行情報であるので、その場合、ステップS505にて先頭行のベースライン位置の設定を行う。先頭行のベースラインは矩形領域の上端から配置を行う文字の文字サイズ分だけ下に下がった位置となる。即ち、

$$\text{baseline} = \text{pos\_y} * \text{配置する文字サイズ} \cdots \cdots \cdots (1)$$

となる。ここで配置する文字サイズは2hに記憶されている書式情報内から取得される。そして、ステップS506にて、文字位置を表す作業変数offsetに1をセットする。

【0051】

ステップS502で $n = 1$ でないと判定された場合には、作成する行情報は領域内の先頭行ではないので、その場合、ステップS503にて、

$$\text{baseline} = \text{baseline} + \text{行ピッチ} \cdots \cdots \cdots (2)$$

によって行のベースラインが設定される。また、文字位置を表す作業変数offsetは、直前行の先頭オフセットstart\_offsetと直前行の行内文字データ数num\_wordsから

$$\text{offset} = \text{start\_offset} + \text{num\_words} \cdots \cdots \cdots (3)$$

として求められる。

【0052】

そして、次に、ステップS507にて、行が領域内に配置できるか否かを判定する。判定はステップS503またはステップS505で算出されたベースラインが領域の下端より上にあるか否かで行う。



【0053】

領域の下端は基準点の y 座標値 `pos_y` に矩形領域の高さ `height` を加算した位置であるから、

$$\text{pos\_y} + \text{height} < \text{baseline} \dots\dots\dots (4)$$

であれば行は配置できない。

【0054】

このステップ S507 で行が配置できないと判断された場合には、ステップ S509 にて行の配置判定変数 `ret` に 0 を代入し、処理を終了する。

【0055】

また、ステップ S507 にて行が配置できると判断された場合には、ステップ S508 にて行の配置判定変数 `ret` に 1 を代入する。その後、ステップ S510 にて、行の先頭文字データオフセットに `offset` を代入する。行の先頭文字データオフセットは領域 3f に記憶される。

【0056】

次に、ステップ S511 にて、行内に配置された文字列の占める長さを表す作業変数 `length` を 0 で初期化する。そして、ステップ S512 にて、領域の幅 `width` が `length` を越えているか否かを判定する。

【0057】

ステップ S512 にて、領域の幅 `width` が `length` を越えていると判定された場合には、ステップ S513 にて、第 `offset` 番目の文字の配置を行う。その時、`length` に第 `offset` 番目の文字データの文字幅を加算する。ステップ S514 にて `offset` の値をインクリメントし、ステップ S515 にて、次の文字があるか否かを判定する。次の文字がなかった場合には、ステップ S515 にて、

$$\text{文字データの個数} < \text{offset} \dots\dots\dots (5)$$

と判定されるので、その場合、ステップ S516 にて行内文字数を設定して処理を終了する。この際、行内文字数 `num_words` は、

$$\text{num\_words} = \text{offset} - \text{start\_offset} - 1 \dots\dots\dots (6)$$

によって求めることができ、領域 3g に記憶される。

## 【0058】

また、ステップS515で、次の文字があると判定された場合には、ステップS512の判定に戻り、行内に文字を配置する領域があるか否かを判定する。

## 【0059】

なお、ステップS512にて、領域の幅widthがlengthを越えない、即ち、offsetの文字を配置すると行から文字が溢れたと判断された場合には、ステップS516にて行内文字数を設定し、処理を終了する。この時の行内文字数は、上に示した式(6)によって求められる。

## 【0060】

以上によって、本実施の形態における文字配置、及びそれに伴う組版情報の設定を行うことができる。

## 【0061】

次に、本実施の形態における溢れ文字の判定方法について、図6に示されるフローチャートに基づいて説明する。

## 【0062】

図6は本実施形態における溢れ文字の判定方法を示すフローチャートである。

## 【0063】

まず、ステップS601にて、組版情報より最終行の行情報を取得する。最終行の行情報は領域3a内の行数num\_linesに基づいて行情報( $n = \text{num\_lines}$ )の領域3cに記憶されている。次に、ステップS602にて、最終行の最終文字データを求める。最終行の最終文字データのオフセットは3fに記憶されている最終行の先頭文字データオフセットstart\_offset及び、3gに記憶されている最終行の行内文字数num\_wordsに基づいて、

$$\text{wCount} = \text{start\_offset} + \text{num\_words} - 1 \dots\dots\dots (7)$$

によって求めることができる。

## 【0064】

次に、ステップS603にて、ステップS602で求めたwCountが領域内の文字データの個数total\_wordsに等しいかどうかを判定する。

## 【0065】

ステップ S603にてwCountが領域内の文字データの個数total\_wordsと等しくないと判定された場合には、配置された最終文字よりも後ろに更に文字データがあることになり、溢れ文字があることになる。その場合には、ステップ S604にて、ret = 1をセットする。

## 【0066】

また、ステップ S603にてwCountが領域内の文字データの個数total\_wordsと等しいと判定された場合には、最後の文字まで配置されていることになるので溢れ文字はない。その場合には、ステップ S605にて、ret = 0をセットする。

## 【0067】

以上により領域内に文字を配置したときに溢れ文字があるか否かの判定ができる。

## 【0068】

図7は文字溢れのある領域、及び文字溢れのない領域を本実施形態に係る装置のCRT1で表示したときの例である。

## 【0069】

図7(A)は文字溢れのない状態での表示方法である。この場合、領域を表す矩形内に対応する文字列（「あいうえお」）が表示されている。

## 【0070】

図7(B)は文字溢れ状態での表示方法である。この場合、領域を表す矩形内に対応する文字列（「あいうえおかきくけこさしすせそたちつてと」）があるが、表示されているのは「あいうえおかきくけこさしすせそ」までであり、文字データ「たちつてと」は表示されていない。また、領域を表す矩形の右下に、溢れ文字があることを示す表示「+」が表示されている。この領域を編集する事によって領域のサイズを変更し、全ての文字が表示された状態が図7(C)である。

## 【0071】

図8及び図9は、領域から溢れた文字列を移動するための手法を表した例である。

## 【0072】

まず、図8では、CRT1上に、図7(A)で示した文字溢れのない状態が表示されている例を示す。この場合、文字を含む矩形領域内にポインタを移動させて、マウスの右ボタンをクリックして領域に対するポップアップメニューを表示しても、溢れ文字が存在しないために、メニュー項目の6番目「溢れた文字列の切り取り」を選択することはできない。

## 【0073】

これに対して、図9では、CRT上に、図7(B)で示した文字溢れ状態が表示されている例を示す。この場合、文字を含む矩形領域内にポインタを移動させて、マウスの右ボタンをクリックして領域に対するポップアップメニューを表示すると、溢れ文字が存在するので、メニュー項目の6番目「溢れた文字列の切り取り」は選択可能であり、この項目を選択すると、実際に、溢れた文字列「たちつてと」の切り取りが実行され、溢れた文字列「たちつてと」は別の領域に移動することになる。

## 【0074】

図10は本実施の形態における領域から溢れた文字列を移動したときのRAM8における記憶データの変化を示した図である。

## 【0075】

切り取りを実行する前の情報から切り取りを実行することによって、領域内に表示されていなかった文字データ「たちつてと」は新たな記憶領域（図10（C））に移動する。この移動された文字列「たちつてと」は任意の文字領域に設定されている文字データ内の任意の位置に挿入することができる。文字領域から移動した元の領域のサイズを拡大した上で元の位置に挿入することも可能である。

## 【0076】

図11は本実施の形態における移動文字データを別の文字データ領域に挿入するときの例である。ポインタをCRT1上の「ABCDE」が表示されている領域に移動させ、さらに、文字列「ABCDE」の「B」と「C」の間を文字カーソルで指示して、マウスの右ボタンをクリックすると、ポップアップメニューが表示される。そこで、そのメニュー項目の5番目「貼り付け」を選択することに

よって、指示した文字カーソルの位置に、切り取られた移動文字列「たちつてと」を挿入することができる。この時、図 12 に示されるように、記憶データ内の文字データは、「ABCDE」から「ABたちつてとCDE」に変更される。

【0077】

文字領域内の文字列が範囲指定されている場合には、挿入ではなく移動された文字列と範囲指定されている文字列とを置換することも可能である。

【0078】

なお、図 11 に示される操作を行わなければ、領域から溢れた文字データは文書内からは削除されたままになる。そして、電源を落とすと、RAM 8 に記憶された溢れ文字データは消去されてしまう。もちろん、電源を落とす際に、RAM 8 から HDD 9 や FDD 10 に溢れ文字データを記憶保存することも可能である。

【0079】

以上の処理を行なうことによって、領域から溢れた文字データの判定と移動処理を行うことが可能となった。

【0080】

【発明の効果】

以上説明したように、本発明によれば、操作者は、文書中の任意の文字領域に対して配置されずに溢れた文字があるか否かの判断を視覚的に行うことができ、また、簡単な操作によって、その文字列を削除したり、別の領域に移動したりすることができるようになる。その結果、操作者が、文書メモリを不用意に浪費することなく、領域から溢れた文字列を簡単な操作で編集できるようになるという効果がある。

【図面の簡単な説明】

【図 1】

本発明の実施の 1 形態に係わる文書作成装置の全体構成を示すブロック図である。

【図 2】

本実施の形態を実現するための文字配置領域を管理するために記憶される情報

を示す図である。

【図 3】

本実施の形態で示される文字編集装置に配置、表示される文字列の配置位置を算出するために使用する組版情報を示す図である。

【図 4】

本実施の形態における領域内行配置処理を行うためのフローチャートである。

【図 5】

本実施の形態における第n番目の行の行情報の作成処理を表すフローチャートである。

【図 6】

本実施の形態における溢れ文字の判定方法を表すフローチャートである。

【図 7】

本実施の形態における溢れ文字表示方法の一例を示す図である。

【図 8】

本実施の形態における領域から文字が溢れていない場合の表示例、および溢れ文字を移動するための手法を示す図である。

【図 9】

本実施の形態における溢れ文字の表示例、および溢れ文字を移動するための手法を示す図である。

【図 10】

本実施の形態における溢れ文字を移動するときの記憶情報の変化を示した図である。

【図 11】

本実施の形態における移動された文字データを別の領域に挿入するための手法を示す図である。

【図 12】

本実施の形態における移動された文字データを別の領域に挿入するときの記憶情報の変化を示した図である。

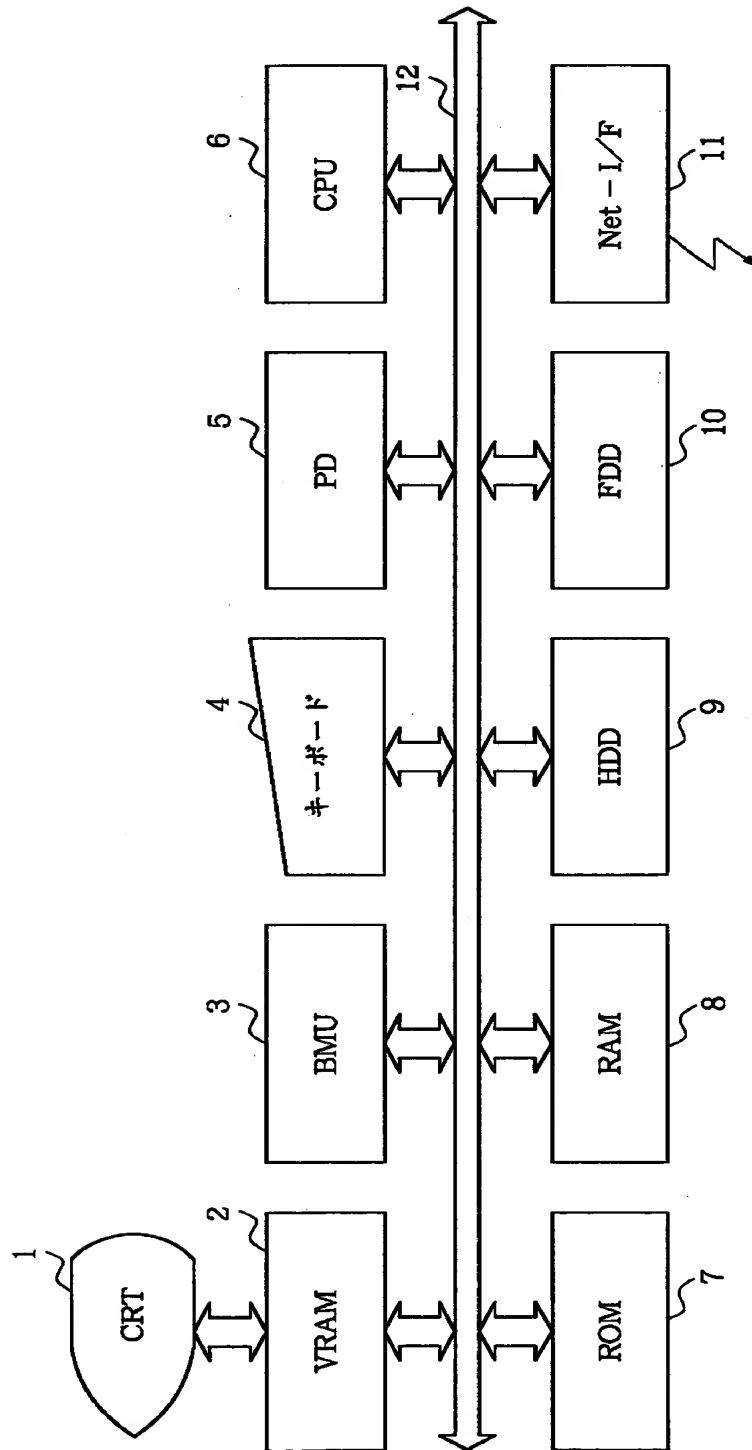
【符号の説明】

- 1 CRT
- 2 VRAM
- 3 BMU
- 4 キーボード
- 5 PD
- 6 CPU
- 7 ROM
- 8 RAM
- 9 HDD
- 10 FDD
- 11 ネットワーク I/F

【書類名】

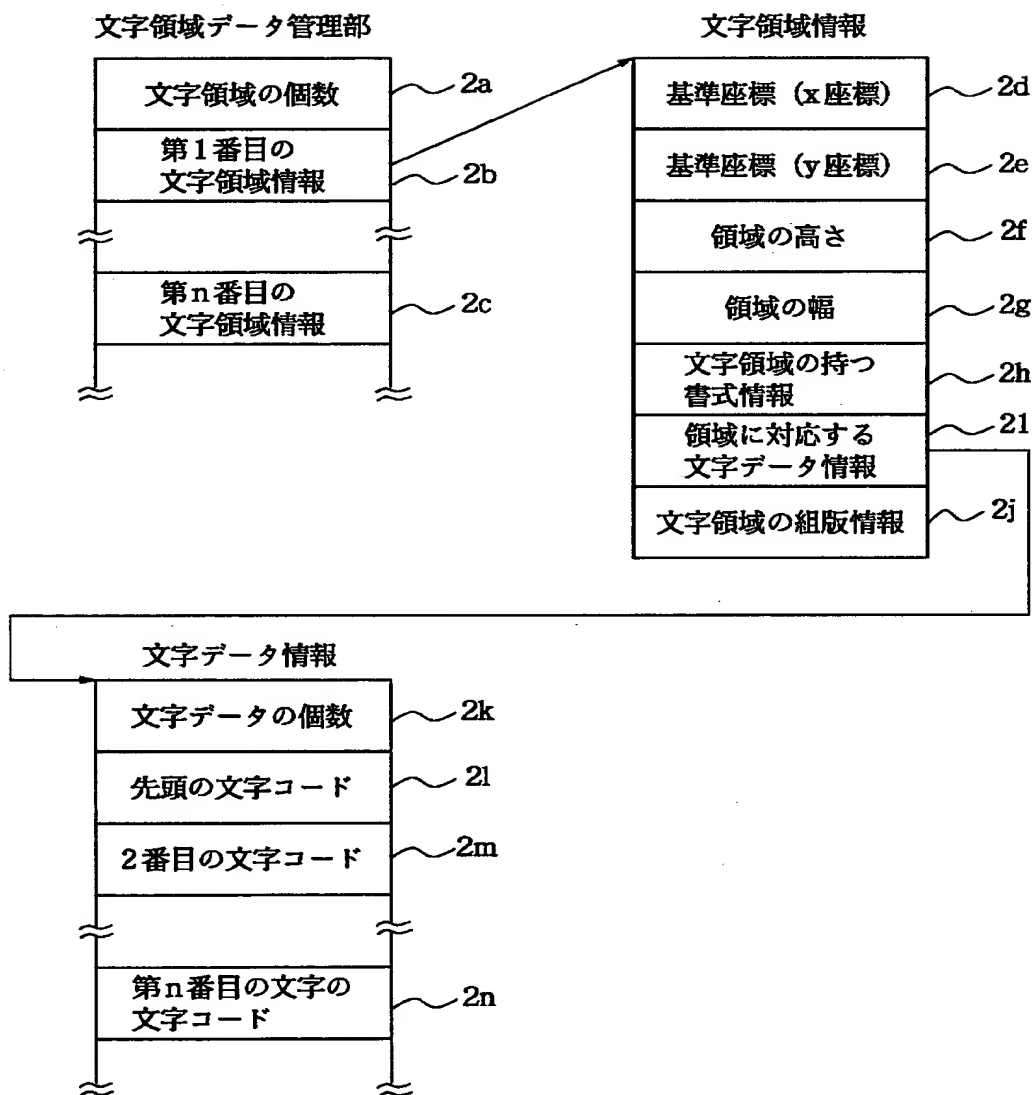
図面

【図 1】

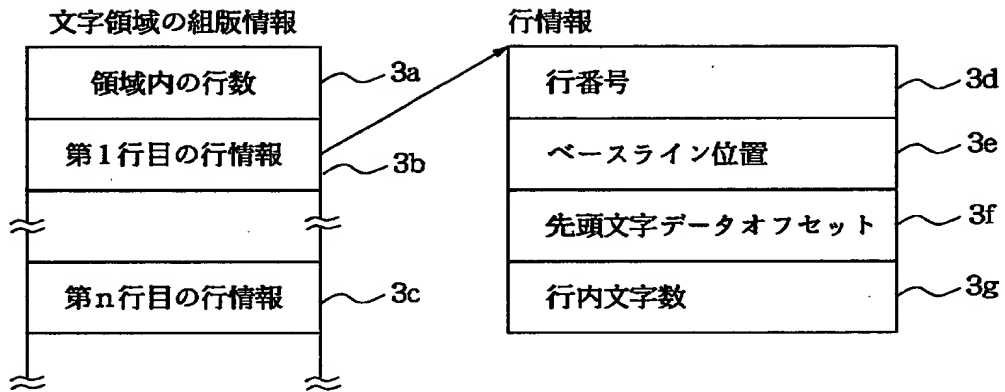




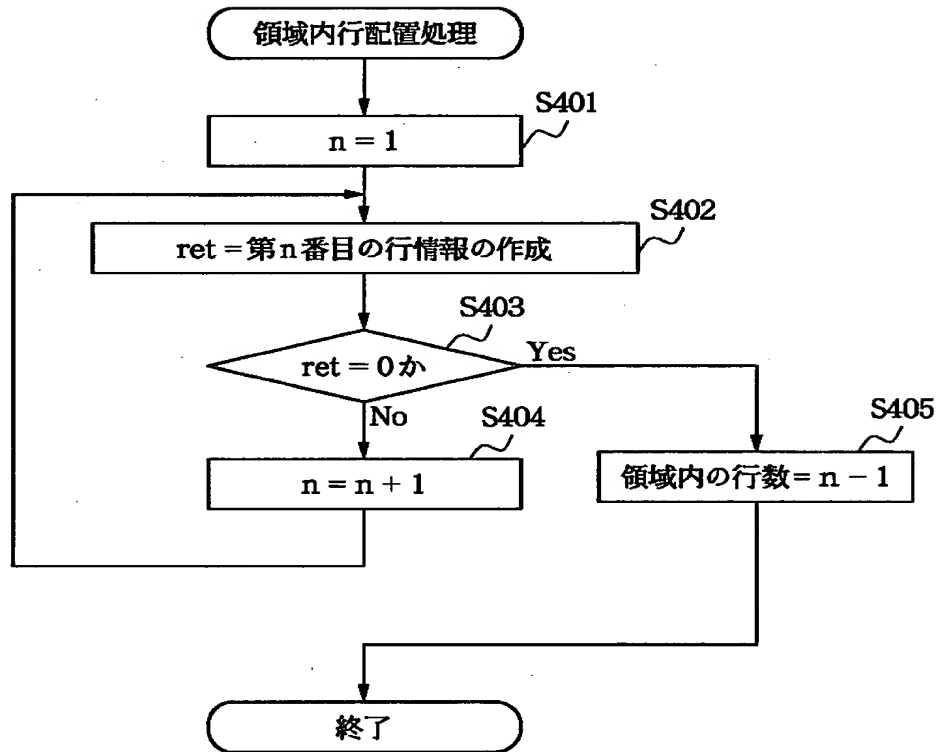
【図 2】



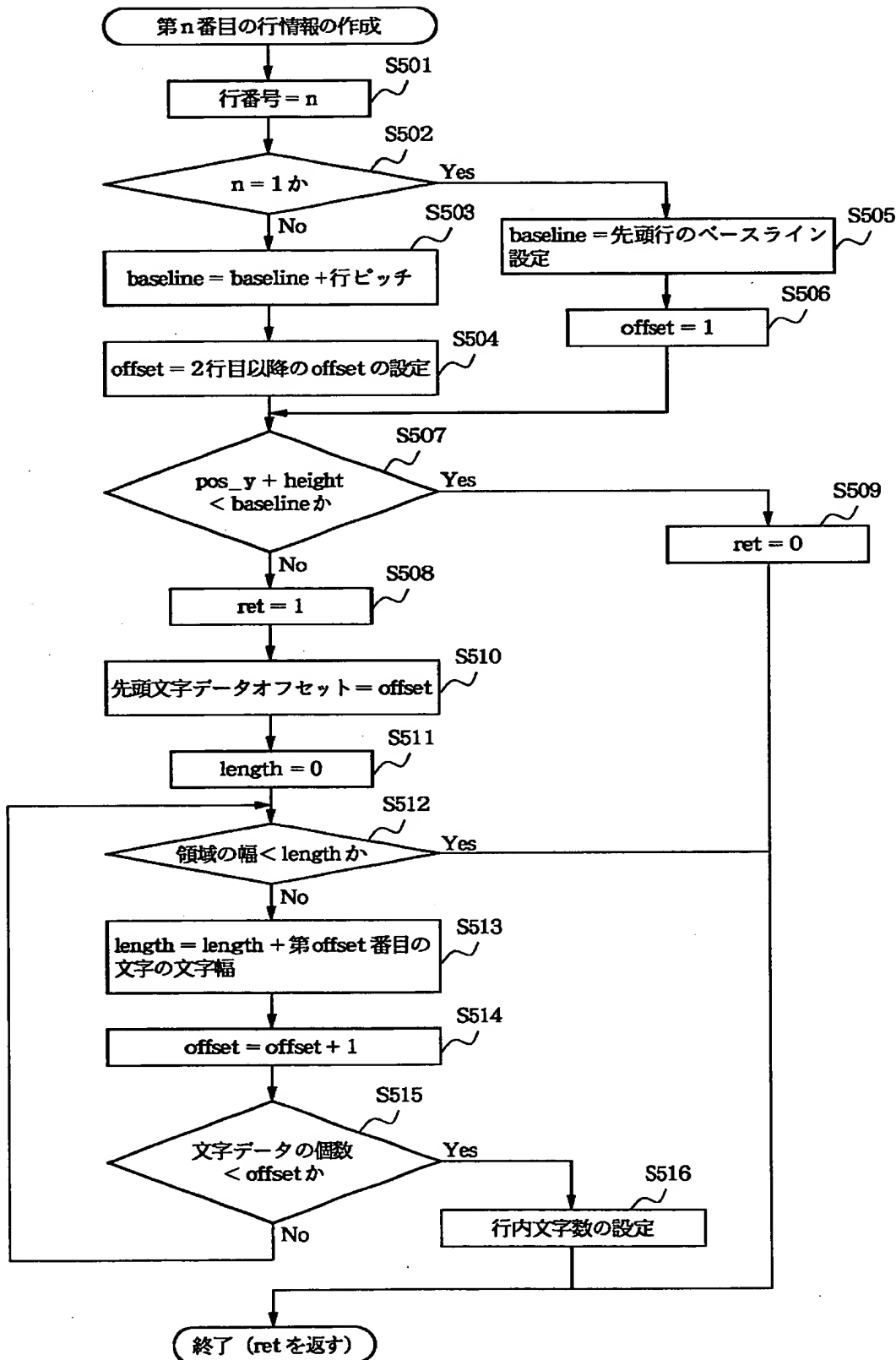
【図 3】



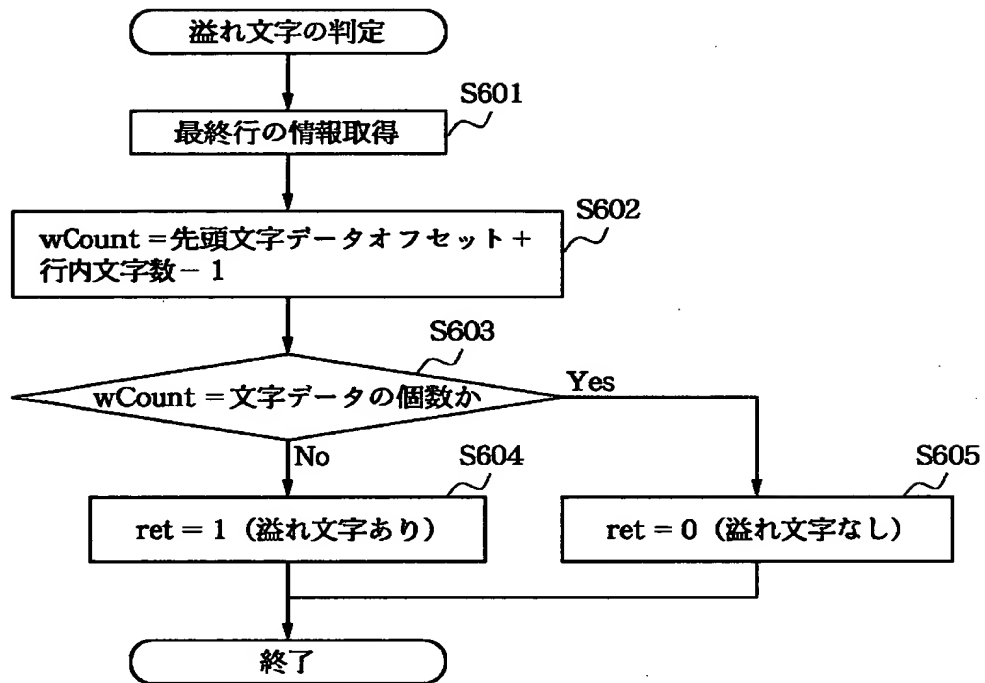
【図 4】



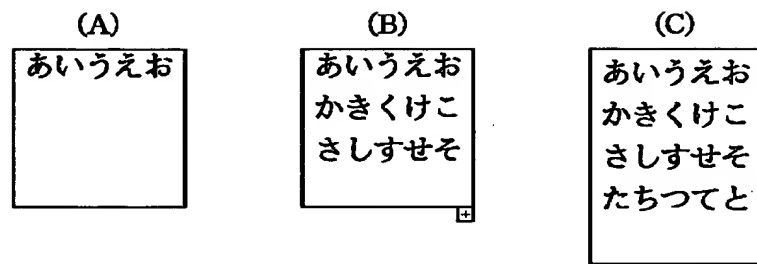
【図 5】



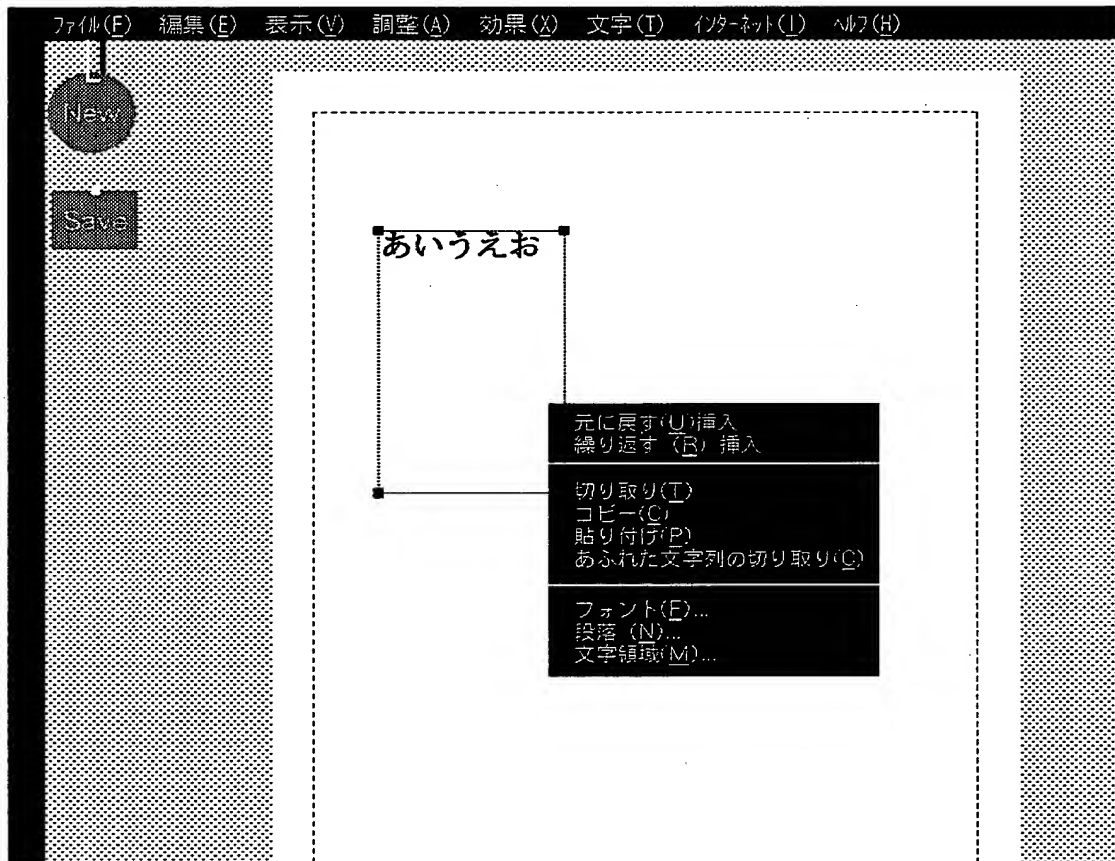
【図 6】



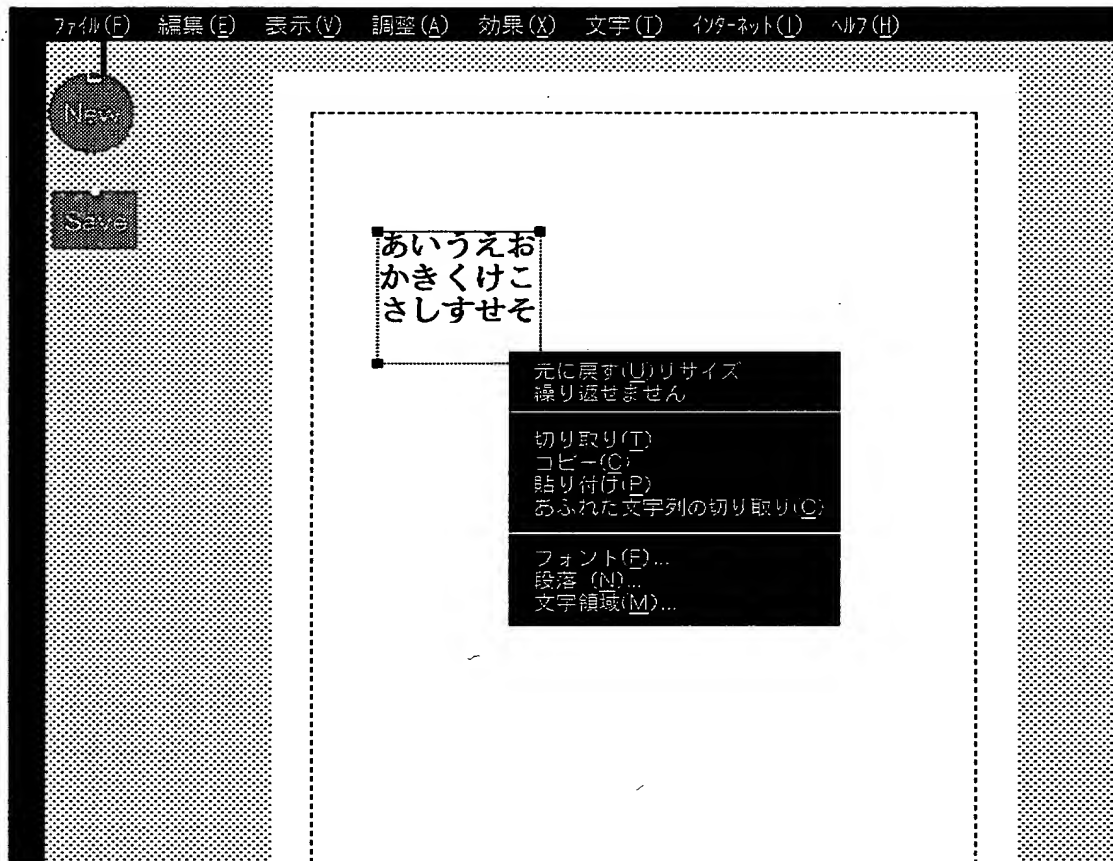
【図 7】



【図 8】



【図9】





【図 10】

(A) 文字データ情報 (切り取り前)

文字データの個数(20)
文字コード あ
文字コード い
文字コード う
文字コード え
文字コード お
文字コード か
文字コード き
文字コード く
文字コード け
文字コード こ
文字コード さ
文字コード し
文字コード す
文字コード せ
文字コード そ
文字コード た
文字コード ち
文字コード つ
文字コード て
文字コード と

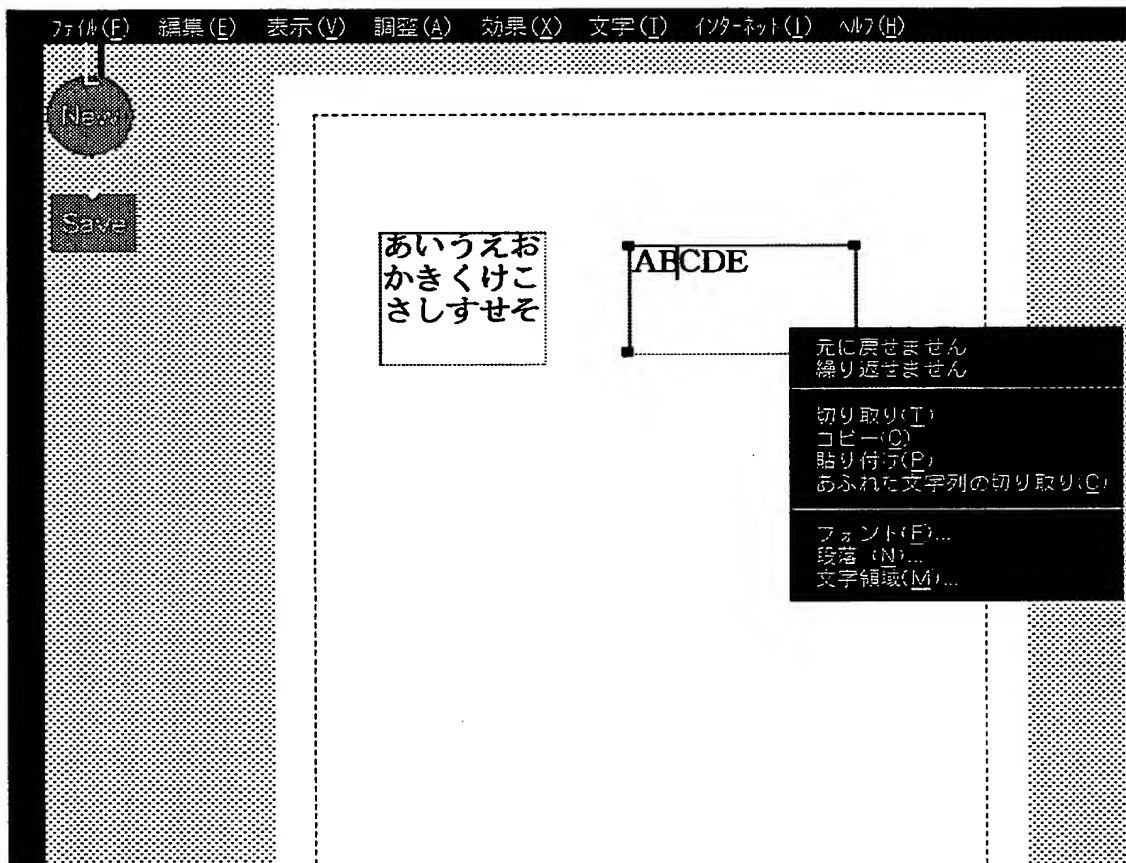
(B) 文字データ情報 (切り取り後)

文字データの個数(15)
文字コード あ
文字コード い
文字コード う
文字コード え
文字コード お
文字コード か
文字コード き
文字コード く
文字コード け
文字コード こ
文字コード さ
文字コード し
文字コード す
文字コード せ
文字コード そ

(C) 移動した文字情報

文字データの個数(5)
文字コード た
文字コード ち
文字コード つ
文字コード て
文字コード と

【図 1 1】



【図 12】

(A) 文字データ情報 (貼り付け前)

文字データの個数(5)
文字コード A
文字コード B
文字コード C
文字コード D
文字コード E

(C) 移動した文字情報

文字データの個数(5)
文字コード た
文字コード ち
文字コード つ
文字コード て
文字コード と

(B) 文字データ情報 (貼り付け後)

文字データの個数(10)
文字コード A
文字コード B
文字コード た
文字コード ち
文字コード つ
文字コード て
文字コード と
文字コード C
文字コード D
文字コード E

【書類名】 要約書

【要約】

【課題】 領域内に配置できない文字データがあることを操作者に認識せしめ、領域内に配置できなかった文字データを別の領域に移動させることによって、領域内に配置できない文字データを容易に編集することができるようにする。

【解決手段】 文字データを配置する領域内に、対応する文字データを一定の規則に従って配置した際に、該領域内に、対応する全ての文字データを配置できたかどうかを判断し、全ての文字データが配置できなかったと判断された場合に、配置できなかった文字データが存在することを識別可能に表示画面上に表示し、配置できなかった文字データを別の記憶領域に移動させて、移動させた文字データを、操作者によって指定された任意の領域に対応する文字データ内に挿入する。

【選択図】 図 11

【書類名】

職権訂正データ

【訂正書類】

特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】

000001007

【住所又は居所】

東京都大田区下丸子3丁目30番2号

【氏名又は名称】

キヤノン株式会社

【代理人】

申請人

【識別番号】

100069877

【住所又は居所】

東京都大田区下丸子3-30-2 キヤノン株式会  
社内

【氏名又は名称】

丸島 儀一

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号

氏 名 キヤノン株式会社